传感器网络分布式数据流的频繁项集挖掘算法

洪月华

（广西大学计算机与电子信息学院 南宁 530004）（广西经济管理干部学院计算机系 南宁 530007）

摘 要 研究无线传感器网络中数据流频繁项集挖掘问题。针对集中式的静态数据流频繁项集挖掘方法不能在传感器网络中直接使用这一特点，提出基于传感器网络的分布式数据流的频繁项集挖掘算法FIMDS。该算法基于FP-tree快速挖掘出传感器节点上单一数据流的局部频繁项集，然后通过路由将其在无线传感器网络里逐层上传合并，在Sink节点上汇聚后，采用自顶向下的高效剪枝策略挖掘出全局频繁项集。实验结果表明，该算法能有效地大幅度减少候选项集，降低无线传感器网络中的通信量，并有较高的时间和空间效率。

关键词 无线传感器网络，分布式数据流，局部频繁项集，全局频繁项集，数据挖掘

中图法分类号 TP311 **文献标识码** A

Frequent Itemsets Mining Algorithm Based on Distributed Data Stream of Sensor Network

HONG Yue-hua

(School of Computer and Electronics and Information, Guangxi University, Nanning 530004, China)

(School of Computer, Guangxi Economic Management Cadre College, Nanning 530007, China)

Abstract This paper mainly studied data stream frequent itemsets mining problem of wireless sensor network. Aiming at the characteristics of sensor networks that centralized static data stream frequent itemset mining method cannot be directly used in sensor network, a frequent itemset mining algorithm FIMDS based on distributed data stream of sensor network was proposed. Based on FP-tree, the algorithm can fast mine the single data stream local frequent Itemsets of sensor nodes, and then through the routing, the local frequent itemsets are uploaded and combined layer-by-layer, and last local frequent itemsets collected on the sink node and global frequent itemsets are got by the top-down efficient pruning strategy. The experimental results show that the algorithm can effectively and greatly reduce candidate itemsets, and reduces the amount of communication traffic in wireless sensor networks, so the algorithm has good performance in time and space.

Keywords Wireless sensor network, Distributed data streams, Local frequent itemsets, Global frequent itemsets, Data mining

1 引言

监测区域中传感器节点收集并同时返回所监测到的信息是无线传感器网络（Wireless Sensor Network，WSN）的基本功能。每个传感器节点持续采集得到的数据是连续的无限数据流。这些数据流具有突发性、实时性和到达高速的特点，而传感器节点本身十分缺乏电池能量、计算能力、存储容量和无线带宽等资源，处理和传输巨大的实时数据流很困难。为了从传感器网络采集得到的海量数据流里获取有用的知识，高效地处理这些数据流就成为新的挑战。能有效解决这个问题的最好方法是对监测得到的感知数据流进行数据挖掘。

数据挖掘研究的一个重要领域是关联规则。关联规则挖掘的核心任务是从数据库中获取频繁项集。长期以来，能够有效挖掘频繁项集的经典算法主要是Apriori及其改进算法[1-4]。然而，这些算法只适用传统静态数据库的单一数据流（即集中式的静态数据流），并需要多次扫描数据库，内存容量高，计算开销大，会产生大量无关候选项集，系统I/O负载大。而无线传感器网络在实际应用环境中，每个传感器监测得到的数据都会产生一个数据流，而多个传感器就形成分布式的多个数据流，同时再加上传感器网络本身资源能量匮乏，现有的集中式的静态数据流频繁项集挖掘方法不能直接在传感器网络中使用。我们必须充分使用传感器节点自身有限的资源能量，来开发适用于无线传感器网络的分布式数据流的频繁项集挖掘算法。

2 算法的问题描述和理论基础

2.1 问题描述

用DB表示无线传感器网络的全局事务数据库，用D表示全局事务条数。设$J_1, J_2, \cdots, J_n$为传感器网络中的节点，$DB_i(i=1,2,\cdots,n)$是传感器节点J_i的单一数据流的局部事

到稿日期：2012-05-21 返修日期：2012-09-23 本文受国家自然科学基金项目（61064002），广西自然科学基金青年项目（2012jjBAG0074），广西教育厅项目（200103YB195）资助。

洪月华（1973—），女，硕士，副教授，主要研究方向为数据挖掘、人工智能与无线传感器网络，E-mail：huibian2005@163.com。

务数据库，D_i 是局部事务条数，则 $DB=\bigcup_{i=1}^{n}DB_i$，$D=\sum_{i=1}^{n}D_i$。

在无线传感器网络中，挖掘全局频繁项集的问题即是如何从 n 个传感器节点收集到的 n 个单一数据流中挖掘出局部频繁项集，各节点间通过无线传感器网络传送有限信息，最终在全局事务数据库 DB 中挖掘出全局频繁项集。

2.2 相关概念、定理和推论

联系关联规则挖掘的有关理论，给出了若干本文算法需要的概念、定理和推论等。

定义 1 项集就是集合的非空子集。项集中元素个数为 k 的项集称为 k-项集。

定义 2 对于某一项集 X，若分别用 $X.count$ 和 $X.count_i$ 表示全局事务数据库 DB 和局部事务数据库 DB_i 中含有 X 的事务条数，则称 $X.count$ 为全局频度，$X.count_i$ 为局部频度。

定义 3 若 X 在 DB 和 DB_i 中的支持度，即 $X.count/D$ 和 $X.count_i/D_i$ 分别用 $X.sup$ 和 $X.sup_i$ 表示，则 $X.sup$ 是 X 在无线传感器网络的全局支持度，$X.sup_i$ 是 X 在传感器节点 J_i 的局部支持度。

定义 4 若项集 X 在无线传感器网络的全局支持度 $X.sup$ 大于等于最小支持度阈值 $minsup$（minimum support thresold），则称 X 是无线传感器网络的全局频繁项集 GFS（Globally Frequent Sets）；而若项集 X 在传感器节点 J_i 的局部支持度 $X.sup_i$ 大于等于最小支持度阈值 $minsup$，则称 X 是传感器节点 J_i 的局部频繁项集 LFS（Locally Frequent Sets）。

定理 1[5] 若项集 X 是传感器节点 J_i 上的局部频繁项集，则 X 的全部非空子集也都是传感器节点 J_i 上的局部频繁项集。类似地，若项集 X 是无线传感器网络上的全局频繁项集，则 X 的全部非空子集也都是该网络上的全局频繁项集。

推论 1[5] 若项集 X 不是传感器节点 J_i 上的局部频繁项集，则 X 的所有超集也一定不是传感器节点 J_i 上的局部频繁项集。类似地，若项集 X 不是无线传感器网络上的全局频繁项集，则 X 的所有超集也一定不是无线传感器网络的全局频繁项集。

定理 2 传感器网络中所有传感器节点的局部频繁项集的并集必定是整个传感器网络全局频繁项集的超集。

定理 3[6] 若项集 X 是无线传感器网络的全局频繁项集，则在该网络中至少存在一个传感器节点 $J_i(1\leqslant i\leqslant n)$，使得项集 X 和它的全部非空子集都是 J_i 的局部频繁项集。

定理 4 若项集 X 不是无线传感器网络中任意一个传感器节点上的局部频繁项集，则 X 一定不是该网络上的全局频繁项集。

为了描述方便，称频繁项集中元素个数为 k 的项集为频繁 k-项集，有两种频繁 k-项集，即局部频繁 k-项集和全局频繁 k-项集；称频繁 1-项集中含有的项目是频繁项目，有两种频繁项目，即局部频繁项目和全局频繁项目。同时用 card (X) 表示集合 X 中含有的元素的个数。

显然，从定理 2 可知，无线传感器网络的全局频繁项集是从各个传感器节点的局部频繁项集中得到的，所以降低在网络中传输局部频繁项集而造成的网络通信开销是提高挖掘效率的关键所在。

2.3 频繁模式树

定义 5 频繁模式树[7]是一种符合如下 3 个要求的树型结构：①它由有且仅有一个特定的标为 null 的结点即树的根（root）、根结点孩子的项目前缀子树集合及为了方便树遍历的频繁项目头表 Htable 这 3 部分组成。②项目前缀子树里的所有结点都是由 item-name（结点名字）、node-count（结点统计）、node-link（结点指针链）和 node-parent（父结点指针链）这 4 个域构成。其中，item-name 表示结点所代表的频繁项目名称，node-count 表示从根结点能到达本结点的路径中所包含结点的交易事务数目，node-link 用来指向频繁模式树中和本结点的项目名称相同的下一个结点，node-parent 指向本结点的父结点。③频繁项目头表 Htable 由 item-name、item-count 和 head of node-link 这 3 个域构成。其中，head of node-link 是指针，用来指向频繁模式树中 item-name 值相同的第一个结点；item-count 用来统计名称为 item-name 的频繁项目值的频度。本文中为了方便叙述，用 FP-tree 表示 DB 对应的频繁模式树，用 FP-tree$_i$ 表示 DB_i 对应的频繁模式树。

上面简要地描述了 FP-tree 的存储结构，如何建立 FP-tree 及基于其的频繁项目集挖掘算法的详细描述可参见文献[7]。

3 分布式数据流的频繁项集挖掘算法 FIMDS

本节具体描述无线传感器网络中分布式数据流的频繁项集挖掘算法 FIMDS（Frequent Items Mining in Distributed Streams），该算法主要由两个子算法组成。子算法 LFSMINING 针对传感器网络中某节点采集得到的单个数据流进行局部频繁项集的挖掘工作。子算法 GFSMINING 中汇聚节点 Sink 通过传感器网络路由接收从各个分布式传感器节点挖掘出的局部频繁项集，合并和交换这些局部频繁项集而形成全局频繁项集的超集，然后经过修剪超集得到精确的全局频繁项集。在无线传感器网络中，分布式数据流的频繁项集挖掘架构如图 1 所示，图中 LFSMINING 是局部频繁项集挖掘算法，GFSMINING 是全局频繁项集挖掘算法。

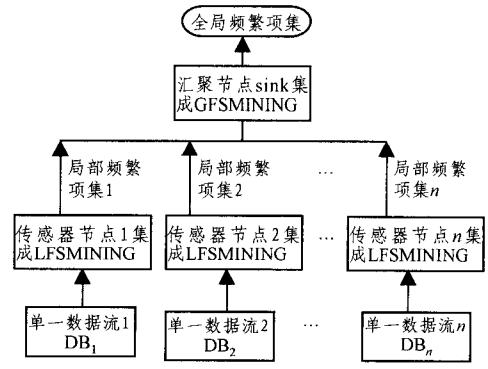

图 1 分布式数据流的频繁项集挖掘架构

3.1 传感器节点上单一数据流的局部频繁项集挖掘算法 LFSMINING

算法 LFSMINING 实现从传感器节点 J_i 采集得到的单一数据流形成的局部事务数据库 $DB_i(i=1,2,\cdots,n)$ 中挖掘出局部频繁项集 LFS$_i$。

J_i 中的任何局部频繁候选项集 X 的支持数就是局部事务数据库 DB_i 中含有 X 的事务数。统计项集的支持数是找

到频繁项集最耗时的计算，占据了全部计算量的大部分工作。本算法在各个传感器节点上的存储结构是使用频繁模式树，算法初始化时，先把 DB_i 压缩存储到频繁模式树 FP-tree$_i$ 中，树中的路径存放相关项目集的频度。根据 FP-tree[7] 的形成原理知道，FP-tree$_i$ 中某唯一一条路径是项集 X 的频繁项目的映射，因此为了大幅度降低计算工作量，可将求 X 在局部事务数据库 DB_i 中的支持数转换为求在频繁模式树 FP-tree$_i$ 中含有 X 的路径数目。这样基于频繁模式树就能确保所有的局部频繁项集是从 FP-tree$_i$ 中挖掘出的，避免了在无线传感器网络里传送大量的项目集，同时也避免了对传感器节点上原始局部事务数据库 DB_i 多次扫描，仅须对 DB_i 扫描一次。本算法完全适应传感器节点在本身资源有限的环境下工作。下面给出该算法的具体描述。

```
LFS_i=∅;
LFCS_i=L_DF={1,2,3,…,k};/* L_DF是频繁项目列表，k=card(L_DF)，LFCS_i是局部频繁候选项集。*/
for all itemsets X={x_1,x_2,…,x_k}∈LFCS_i{
  LFCS_i=LFCS_i-X;
  If X不是LFS_i中某元素的子集
  {call supcount(FP-tree_i,X,Htable,LFCS_i);/*项集X在DB_i中的
   支持数X.cout_i通过调用函数计算得到*/
  X.sup_i=X.cout_i/D_i
  If X.sup_i≥minsup then LFS_i=LFS_i∪X;
  Else for all item x∈X /*X的所有k-1项集加入LFCS_i*/
  If (X-{x}不是LFS_i或LFCS_i中任一子集)then
    LFCS_i=LFCS_i∪{X-{x}};
  }
}
```

函数 supcount 是计算局部频繁候选项集 LFCS$_i$ 里的项集 X 在局部事务数据库 DB_i 中的支持数 $X.cout_i$。算法不是基于原始局部事务数据库 DB_i，而是根据频繁模式树 FP-tree$_i$ 完成支持数统计的。因为 LFCS$_i$ 中的所有项目集中的项目都是按照其支持数的降序排列，所以判断含有项集 X 的路径数是相当容易的，只需频繁模式树 FP-tree$_i$ 的父指针遍历根节点而无须遍历叶节点。函数具体算法描述如下：

```
Procedure supcount(tree,X,Htable,LFCS_i)
{
  查找项目头表Htable的item-name，找到Htable[X].term-name的
  值为i;
  在tree中依照Htable[X].head找出名称为i的节点d_1,d_2,…,d_q;
  依照节点d_1,d_2,…,d_q和它们的前缀节点的父节点指针域找出含
  有i的全部路径p_1,p_2,…,p_q;
  For all X∈LFCS_i do
  For (k=1;k<=q;k++)do
    If (路径p_k含有X) then X的支持数X.cout_i增加;
}
```

3.2 Sink 节点上全局频繁项集挖掘算法 GFSMINING

假设无线传感器网络中有 n 个传感器节点，每个节点监测得到的单一数据流是 S_i $(1\leqslant i\leqslant n)$。$n$ 个传感器节点分别用本文算法 LFSMINING 挖掘出 DB_i 的局部频繁项集 LFS$_i$，通过路由将其在无线传感器网络里逐层上传合并后汇聚在 Sink 节点上，然后同时使用自顶向下高效剪枝策略进行候选项集的剪枝工作来挖掘出全局频繁项集 GFS。

由定理 2 和定理 3 知，全局频繁项集肯定是某个局部频繁项集的子集，各传感器节点挖掘出的局部频繁项集的并集一定是整个传感器网络全局频繁项集的超集，Sink 节点上全局频繁项集挖掘算法工作流程如下：

Step 1　Sink 节点向各个传感器节点广播网络初始化信息。

Step 2　根据各个传感器节点返回的消息建立无线传感器网络路由树。

Step 3　Sink 节点向各个传感器节点广播诸如最小支持度等关联规则挖掘参数。

Step 4　建立各个传感器节点 J_i 的 FP-tree$_i$。

```
For(i=1;i<=n;i++)
{Scan DB_i once;
  Collect the set of local items F_i and their supports;
  Sink collects global frequent items F from all F_i;
}
L_DF is sorted in the order of descending support count;/*频繁项目列
表L_DF由支持数降序排列得到*/
Sink sends L_DF to other J_i;
Creat an FP-tree_i/*用文献[8]生成节点J_i的FP-tree_i*/
```

Step 5　各个传感器节点利用本身有限的资源对收集得到的单一数据流进行局部频繁项集挖掘(算法 LFSMINING)。

```
For(i=1;i<=n;i++)
  Call LFSMINING(FP-tree_i,J_i,minsup,LFS_i);
```

Step 5　各个传感器节点广播挖掘出的局部频繁项集，同时通过路由将其发送到 sink 节点。

```
For(i=1;i<=n;i++)
  J_i sends LFS_i to sink;
```

Step 6　sink 节点将收到的各个传感器节点的局部频繁项集进行合并运算 $\bigcup_{i=1}^{n}LFS_i$ 得到 GFCS。

```
Sink combines LFS_i and produces GFCS
```

Step 7　在 sink 节点上对全局频繁项候选集 GFCS 进行自顶向下剪枝，这种剪枝方法不会产生全局频繁项集的任何超集，也不会在传感器网络中传输非全局频繁项集的超集，由此网络通信开销得到大幅度降低，从而挖掘出所有全局频繁项集 GFS。

```
GFS=∅
While(GFCS≠∅)
{ k=max{card(x)|x∈GFCS} /*k是全局频繁项候选集GFCS中所
  有项集的最大项数*/
For all 项集X∈k-项集 in GFCS
If X不是GFS中所有项集的子集
{ sink broadcast X;
  For(i=1;i<=n;i++)
    { 传感器节点J_i sends X.cout_i to sink;/*项集X的局部频度X.
      cout_i是从传感器节点J_i的局部FP-tree_i的各路径中快速计算
      得到*/
  X.cout=∑_{i=1}^{n} X.count_i
  }
X.sup=X.cout/D;
If X.sup <minsup/*X不是全局频繁项集*/
{ sink deletes X from GFCS;
  For all item x∈X do
    If(X-{x}不是GFS中任一子集) then
      GFCS=GFCS∪{X-{x}}/*将项集X的所有k-1项子集加
           入GFCS*/
```

(下转第 94 页)

[J]. 信息与控制,2006,35(2):129-134
[5] Okdem S, Karaboga D. Routing in wireless sensor networks using ant colony optimization [C]// First NASE/ESA Conference on Adaptive Hardware and System-AHS. 2006:401-404
[6] 任秀丽,梁红伟,汪宇. 基于多路径蚁群算法的无线传感器网络的路由[J]. 计算机科学,2009,36(4):116-118
[7] Dorigo M, Stützle T. Ant Colony Optimization[M]. MIT Press: Cambridge, MA, USA, 2004
[8] Heinzelman W B, Chandrakasan A P, Balakrishnan H. An Application-Specific Protocol Architecture for Wireless Microsensor Networks [J]. IEEE Trans. on Wireless Communications, 2002,1:660-670
[9] Tian Y, Wang Y, Zhang S. A novel chain-cluster based routing protocol for wireless sensor networks [C]// International Conference on Wireless Communications, Networking and Mobile Computing. 2007:2456-2459
[10] Manjeshwar A, Agrawal D P. TEEN: A routing protocol for enhanced efficiency in wireless sensor networks [C]// 15th International Parallel and Distributed Symposium. San Francisco, CA, USA, 2001:2009-2015
[11] Dai Z, Li Z, Wang B, et al. An Energy-Aware Cluster-Based Routing Protocol for Wireless Sensor and Actor Network [J]. Information Technology, 2009, 8(7):1044-1048
[12] Krishnamachari B, Estrin D, Wicker S. Modelling data-centric routing in wireless sensor networks [C]// Proc. of the IEEE Infocom. 2002:2-14

(上接第 60 页)

```
}
else/ * X 是全局频繁项集 * /
  { GFS=GFSU{X}; / * 在全局频繁项集 GFS 中加入 k-项集 X * /
    Sink deletes 项集 X 及其所有非空子集 from GFCS; / * 从全局频
    繁项候选集 GFCS 中把项集 X 和它的所有非空子集全部删除 * /
  }}}
```

4 算法实现与性能分析

本文算法 FIMDS 和对比算法 PLT-STREAM[9] 是在一台式机上用 VC++编程实现的,无线传感器网络环境的模拟通过多线程进行,实验数据来自 Intel Berkeley Research lab[10] 中的 10 天历史数据。运行算法的台式机配置如下:CPU 为 Intel Pentium Ⅳ 3.0GHz,内存为 2GB DDR2,操作系统为 Windows XP。

图 2 和图 3 是在不同的支持度下两个算法运行 10 天的数据时所占用的 CPU 时间和内存的比较。从图中看到,算法 FIMDS 运行时间较快,占用内存空间较少。由于 PLT-STREAM 算法扫描数据库两次,而算法 FIMDS 只扫描一次,因此极大地缩减了 CPU 的运行时间和内存的使用量。

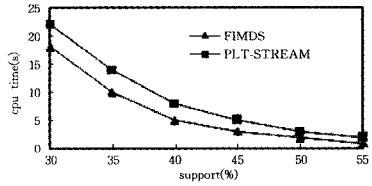

图 2 运行时间对比

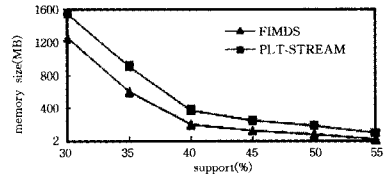

图 3 内存使用对比

图 4 说明两种算法在无线传感器网络中的传输通信量在刚开始阶段都随着传感器节点中数据流量的增大而增大。不过算法 FIMDS 的传输通信量增长相对缓慢,特别是随着数据流量增大到一定程度,该算法逐渐找出频繁项集后,就再也不会在传感器网络中传输非全局频繁项集的超集,使得通信量趋于稳定,而算法 PLT-STREAM 对于已经挖掘出的全局频繁项集的子集仍然还在传感器网络中传输,不能有效缩减项目集的传输数,随着数据流量的增大,通信量大大高于算法 FIMDS。

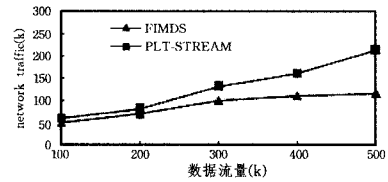

图 4 网络传输通信量对比

结束语 本文提出能快速挖掘出无线传感器网络中分布式数据流的频繁项集的算法 FIMDS。该算法采用的存储结构是频繁模式树,项目集的频度能迅速地从各局部频繁模式树 FP-treei 的有关路径中获得,对高速到达的数据流只需扫描一遍数据库,使用自顶向下的剪枝策略能有效地减少全局频繁候选项集的生成和传输,网络通信消耗得到降低,运行效率较高。

参 考 文 献

[1] Agrawal R, ImielinSki T, Swami A. Mining association rules between sets of items in large database[C]// The ACM SIGMOD International Conf on Management of Data. Washington, DC, 1993:207-216
[2] 蔡伟贤,滕少华. 改进的 Apriori_TFP 算法入侵检测中的应用[J]. 计算机工程与设计,2011,32(11):3594-3598
[3] 刘维晓,陈俊丽,屈世富,等. 一种改进的 Apriori 算法[J]. 计算机工程与应用,2011,47(11):149-151
[4] 张浩,景凤宣,谢晓尧. 基于数据挖掘关联规则 Apriori 改进算法的入侵检测系统的研究[J]. 贵州师范大学学报:自然科学版,2011,29(3):84-87
[5] 何波,王华秋,刘贞. 快速挖掘频繁项集的并行算法[J]. 计算机应用,2006,26(2):391-392
[6] 陆介平,杨明,孙志辉. 快速挖掘全局最大频繁项目集[J]. 软件学报,2005,16(4):553-560
[7] Han J W, Pei J, Yin Y. Mining frequent patterns without candidate generation[C]// Proc of the 2000 ACM SIGMOD Int Conf on Management of Data. Dallas: ACM Press, 2000:1-12
[8] 杨明,孙志挥,吉根林. 快速挖掘全局频繁项目集[J]. 计算机研究与发展,2003,40(4):620-626
[9] 梅淑英,林亚平,周四望,等. 一种基于字典树的传感器节点关联规则的挖掘算法[J]. 计算机工程与科学,2010,32(4):119-124
[10] Intel Lab Data [EB/OL]. http://berkeley.intel-research.net/labdata/,2008-08-30
[11] 王茜,张鲲鹏. 隐私保护数据挖掘算法 MASK 的改进[J]. 重庆理工大学学报:自然科学版,2012,26(6):63-66